

RasPi

DESIGN
BUILD
CODE

31



MASTER
SENSE
HAT
SKILLS

MAKE A PI GAME IN MINECRAFT

Plus
Anonymise
your web browsing



Welcome



Minecraft is such a great resource for learning how to code. It's approachable, fun and much more flexible than you might think. This issue, we explain how you can use it to create your very own version of the classic puzzler *Minesweeper*. That's not the only game we're looking at this issue either; you'll also learn how to start coding a version of *Rock, Paper, Scissors, Lizard, Spock* using the Sense HAT. Plus we take a look at an incredible Pi-powered RC submersible, demonstrate how you can use the Raspberry Pi and Python to set up a motion tracker, explain how to use Fritzing to design your own circuit diagrams, and anonymise your internet browsing using the Pi and TOR. Enjoy!

April

Editor

From the makers of
Linux User
& Developer

Join the conversation at...

 @linuxusermag

 Linux User & Developer

 RasPi@imagine-publishing.co.uk

Get inspired

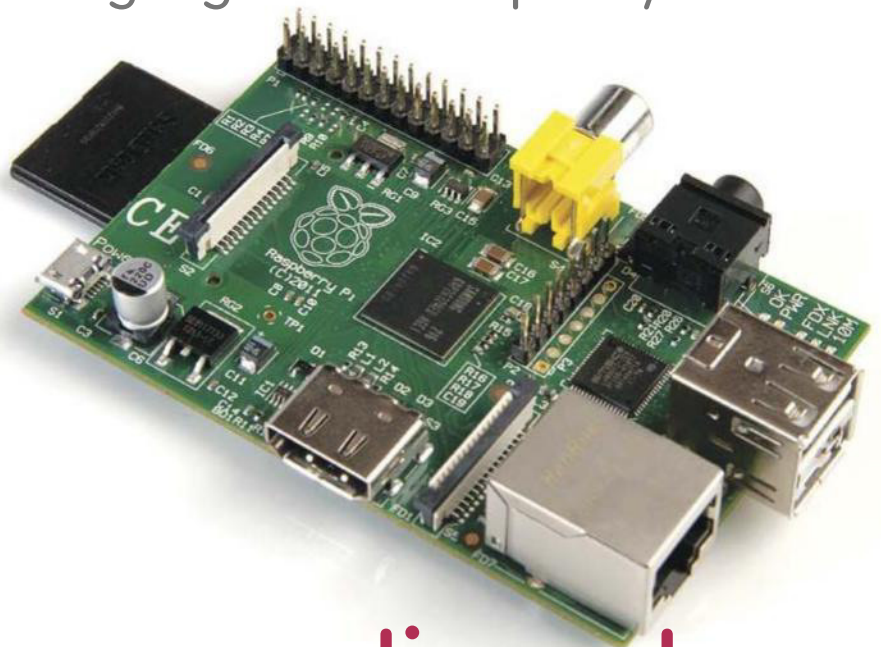
Discover the RasPi community's best projects

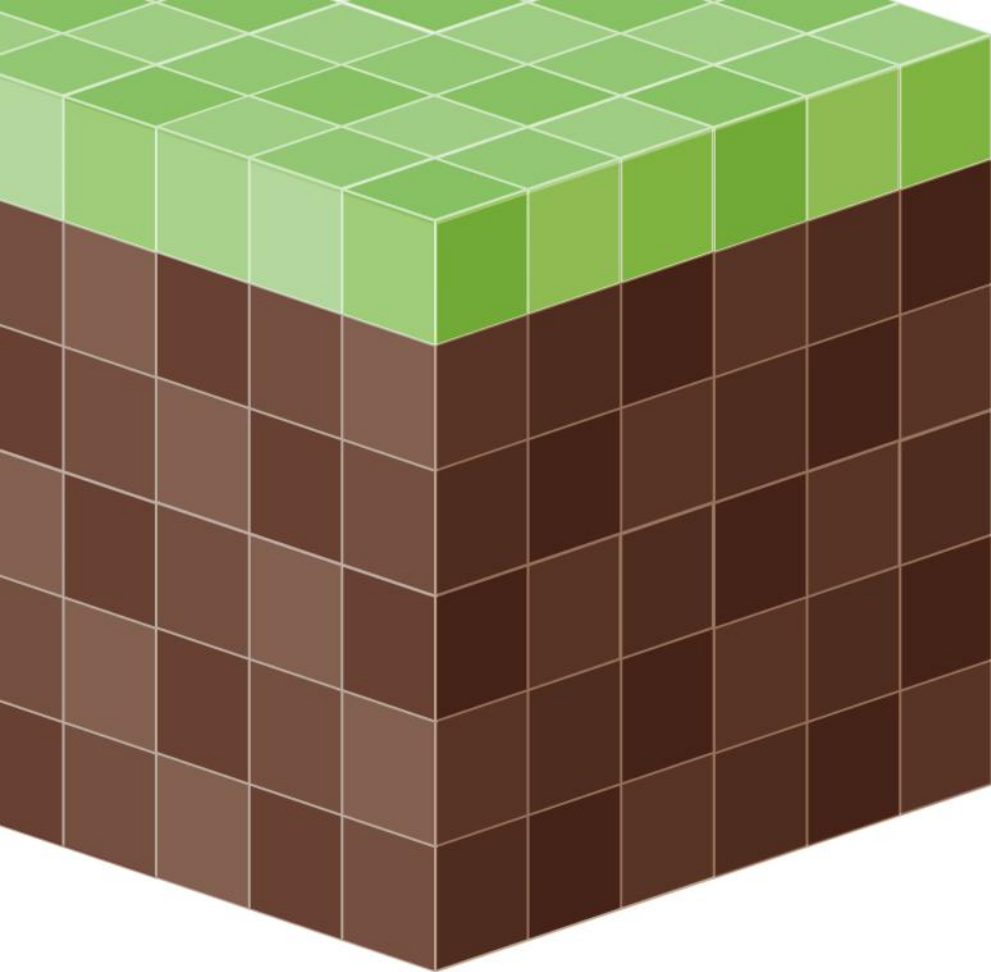
Expert advice

Got a question? Get in touch and we'll give you a hand

Easy-to-follow guides

Learn to make and code gadgets with Raspberry Pi





Contents

Create a Minecraft Minesweeper game

Use your Raspberry Pi and Python knowledge to code a game



Pi project: Underwater Pi drone

Niels Affourtit has turned his passion for RC vehicles into a sub



Master essential Sense HAT skills

Start a coding a version of *Rock, Paper, Scissors, Lizard, Spock*



Anonymise your web traffic

Use your Raspberry Pi as an anonymous wireless access point



Create your own circuit diagrams with Fritzing

Illustrate circuit diagrams with this handy utility



Motion tracking with your Pi

Learn how to track motions with your Raspberry Pi



Talking Pi

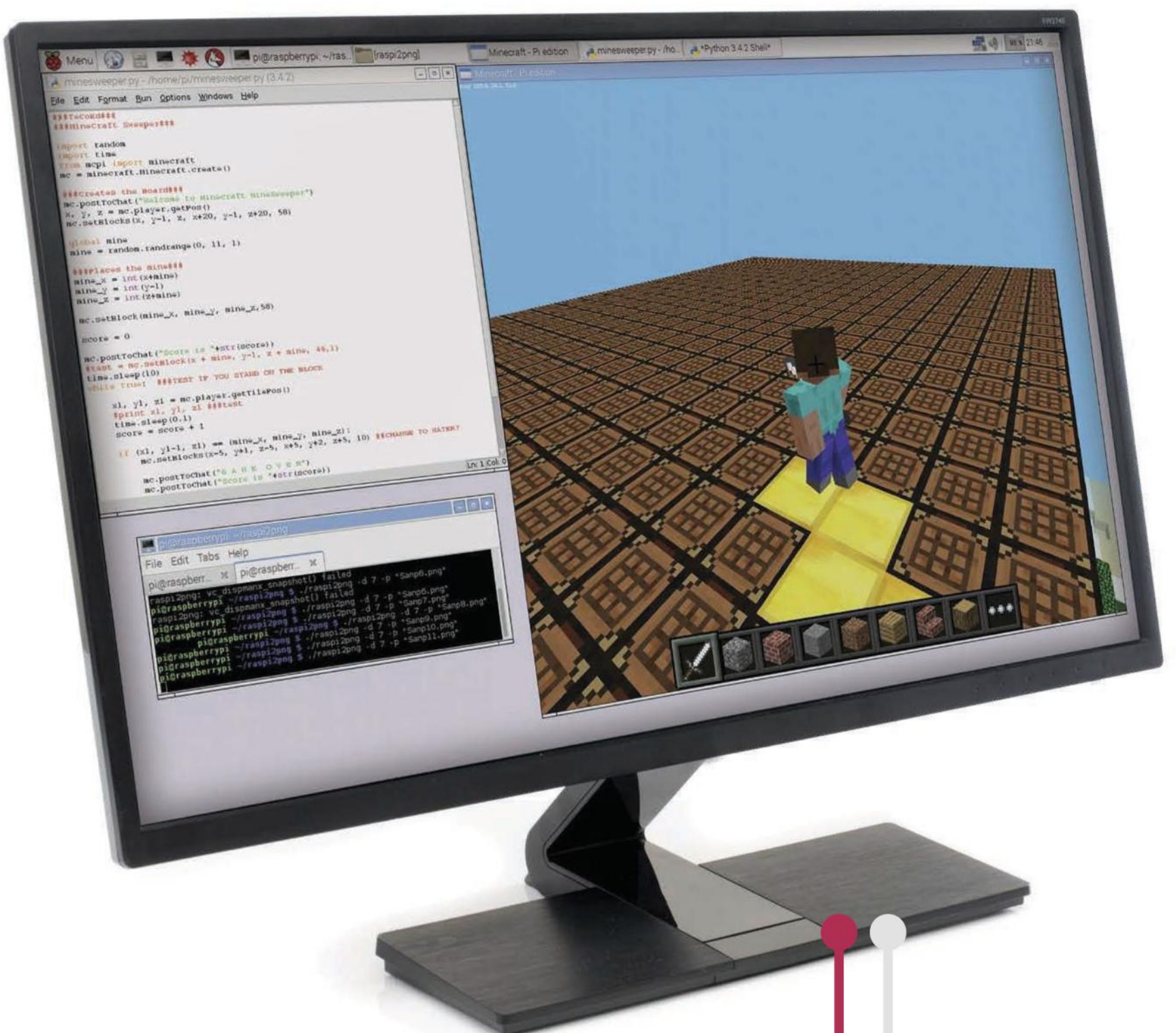
Your questions answered and your opinions shared

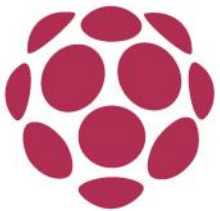




Create a Minecraft Minesweeper game

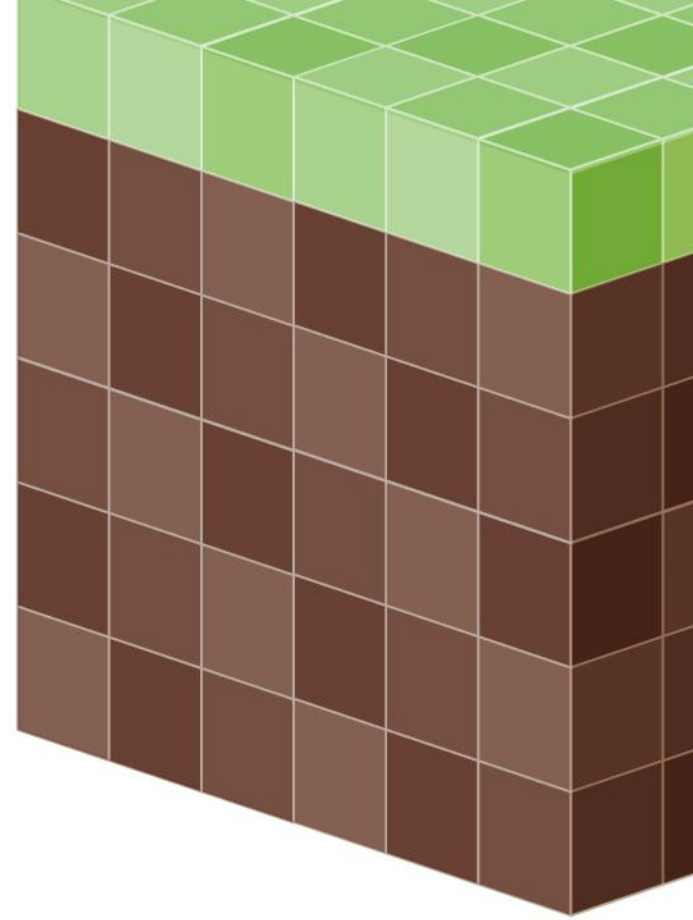
Use your Raspberry Pi and Python knowledge to code a simple mini-game in Minecraft





You may remember or have even played the classic *Minesweeper* PC game that originally dates back to the 60s. Over the years it has been bundled with most operating systems, appeared on mobile phones, and even featured as a mini-game variation on *Super Mario Bros*.

This project will walk you through how to create a simple version in *Minecraft*: it's *Minecraft Minesweeper*! You will code a program that sets out an arena of blocks and turns one of these blocks into a mine. To play the game, guide your player around the board. Each time you stand on a block you turn it to gold and collect points, but watch out for the mine as it will end the game and cover you in lava!



01 Update and install

To update your Raspberry Pi, open the terminal and type:

```
sudo apt-get upgrade  
sudo apt-get update
```

The new Raspberry Pi OS image already has *Minecraft* and Python installed. The Minecraft API which enables you to interact with *Minecraft* using Python is also pre-installed. If you are using an old OS version, it will be worth downloading and updating to either the Jessie or Raspbian image downloadable here: www.raspberrypi.org/downloads.

02 Importing the modules

Load up your preferred Python editor and start a new window. You need to import the following



for it to load – as you walk around, you'll drop flowers! Change the ID number in line 1 to change the block type, so instead of flowers, try planting gold, water or even melons.

05 Posting a message to the Minecraft world

It is also possible to post messages to the Minecraft world. This is used later in the game to keep the player informed that the game has started and also of their current score. In your previous program add the following line of code under the flower = 38 line, making this line 2: `mc.postToChat("I grew some flowers with code")`. Now save and run the program by pressing F5 – you will see the message pop up. You can try changing your message, or move to the next step to start the game.

06 Create the board

The game takes place on a board created where the player is currently standing, so it is advisable to fly into the air or find a space with flat terrain before running your final program. To create the board you need to find the player's current location in the world using the code `x, y, z = mc.player.getPos()`. Then use the `mc.setBlocks` code in order to place the blocks which make up the board:

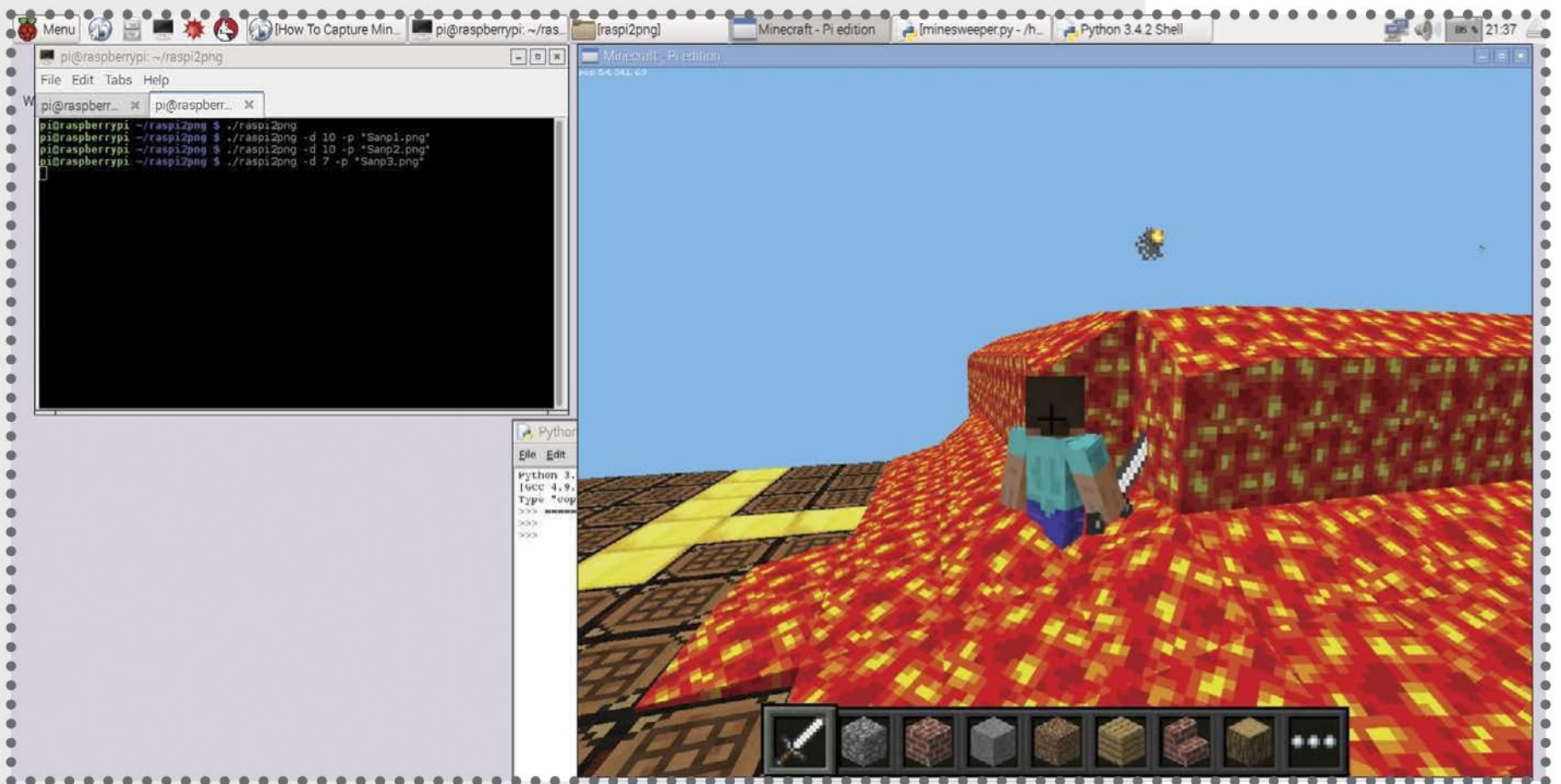
```
mc.setBlocks(x, y-1, z, x+20, y-1, z+20, 58).
```

The number 58 is the ID of the block that is a crafting table. You can increase or decrease the size of the board by changing the +20. In the code example

Switching to the shell

Switching between the Python Shell and Minecraft window can be frustrating, especially as MC overlays the Python window. The best solution is to half the windows across the screen. (Don't run MC full-screen as the mouse coordinates are off). Use the Tab key to release the keyboard and mouse from the MC window.





above, the board size is 20 x 20 blocks, which gives you a 400-block arena to play within.

Right Nothing says “game over” quite like a huge eruption of lava

07 Creating the mine

In the previous step you found the player’s location on the board. This x, y, z data can be reused to place the mine on the board. The code `mine = random.randrange(0, 11, 1)` generates a random number between 1 and 10. Combine this with the player’s current x axis position and add the random number to the position – this creates a random mine block on the board.

```
mine_x = int(x+mine)
mine_y = int(y-1)
mine_z = int(z+mine)
```

Use `setBlock` to place the mine: `mc.setBlock(mine_x, mine_y, mine_z, 58)`. Using `y-1` ensures that the block is placed on the same level as the board and is therefore

hidden. The number 58 is the block ID, which you can change if you wish to see where the mine is; this is useful for testing that the rest of the code is working correctly. Remember to change it back before you play!

08 Create a score variable

Each second that you remain alive within the game, a point is added to your score. Create a variable to store the current score, setting it to a value of zero at the beginning of the game. Use the `postToChat` code to announce the score at the beginning of the game. Note that MC cannot print a value to chat, so the score is first converted into a string before it is displayed.

```
score = 0
mc.postToChat("Score is "+str(score))
time.sleep(10)
```

09 Check the player's position on the board

So far you have created a board that includes a randomly-placed mine the same colour as the board so you can't see it! Next, you need to check the player's position on the board and see if they are standing on the mine. This uses a while loop to continually check that your player's position is safe, no mine, else game over. Since the player's coordinate position is used to build the original board and place the mine, you have to find the player's position again and store it as a new variable – `x1`, `y1` and `z1` – otherwise the board shifts around as the player moves.

```
while True:
```

Other Minecraft hacks

If you enjoy programming and manipulating Minecraft then there are more great Raspberry Pi -based projects for you to check out. Our expert has a bunch of them at tecoed.co.uk/minecraft.html. The folks behind Adventures In Minecraft have some great guides over at stuffaboutcode.com/p/minecraft.html, as well.




```
x1, y1, z1 = mc.player.getTilePos()
```

10 One point, please

Now that the player has moved one square they are awarded a point. This is a simple action of adding the value one to the existing score value. This is achieved using `score = score + 1`. Since it sits inside a loop, it will add one point each time the player moves.

```
time.sleep(0.1)
```

```
score = score + 1
```

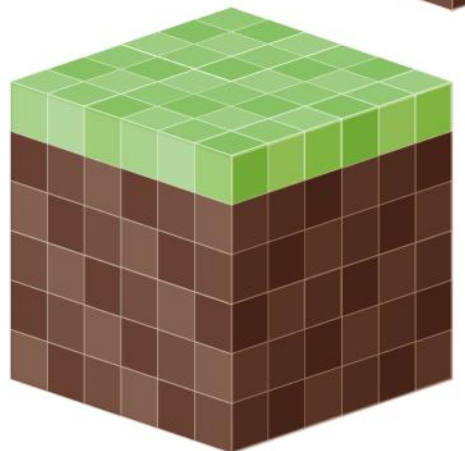
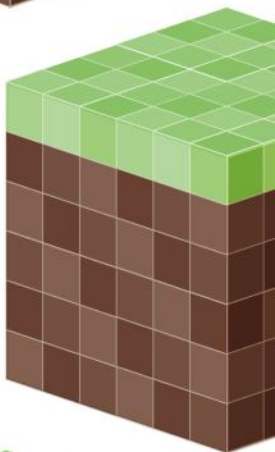
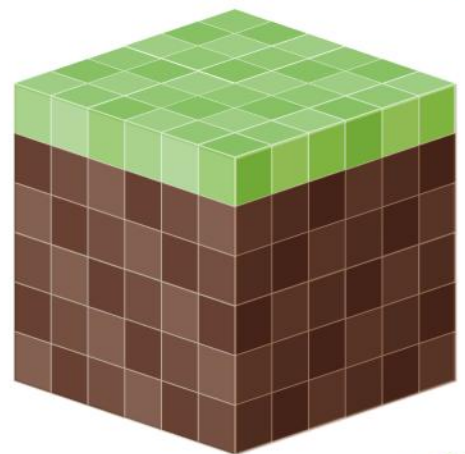
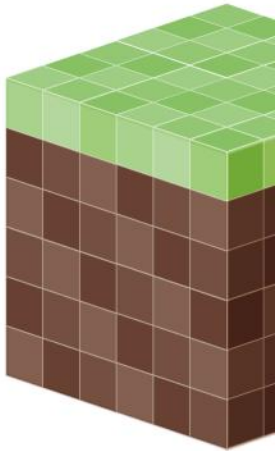
11 The tension increases...

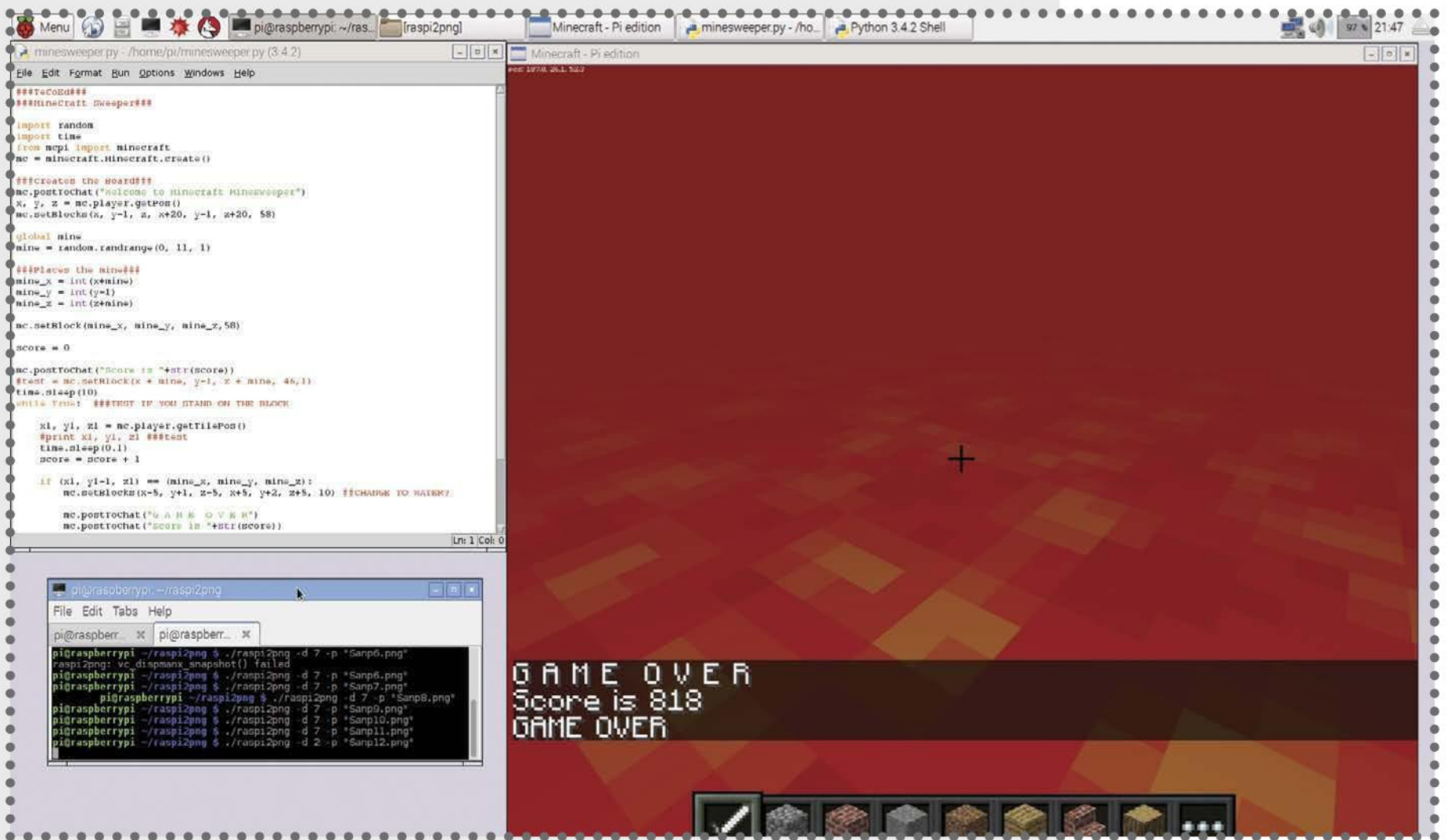
Once you have been awarded the point, the next stage of the game is to check whether the block you are standing on is a safe block or if it is the mine. This uses a conditional to compare the coordinates of the block beneath you – `x1, y1-1, z1` – with the `mine_x, mine_y, mine_z` position of the mine. If they are equal then you are standing on the mine. In the next step you will code the explosion:

```
if (x1, y1-1, z1) == (mine_x, mine_y, mine_z):
```

12 Setting the mine off

In the previous step a conditional checks whether you are standing on the mine or a safe block. If it is the mine then it will explode. To create this, use lava blocks, which will flow and engulf the player. You can use the `mc.setBlocks` code to set blocks between two points. Lava blocks are affected by gravity, so setting them





higher than the player means that the lava flows down over the player.

```
mc.setBlocks(x-5, y+1, z-5, x+5, y+2, z+5,
10)
```

13 Game over

If you do stand on the mine, the game is over.

Use the post to chat code to display a "Game Over" message in the Minecraft World.

```
mc.postToChat("G A M E O V E R")
```

14 Final score

The last part of the game is to give a score. This uses the score variable that you created in Step 8 and then uses the `mc.postToChat` code. Convert the score

to a string first so that it can be printed on the screen. Since your turn has ended, add a break statement to end the loop and stop the code from running.

15 Safe block

But what if you

But what if you missed the mine? The game continues and you'll need to know where you have previously been on the board. Use the code `mc.setBlock(x1, y1-1, z1, 41)` to change the block you are standing on into gold or another material of your choice. In the code, the Y position is $Y - 1$, which selects the block beneath the player's feet.

16 Increment the score

As well as living to play and

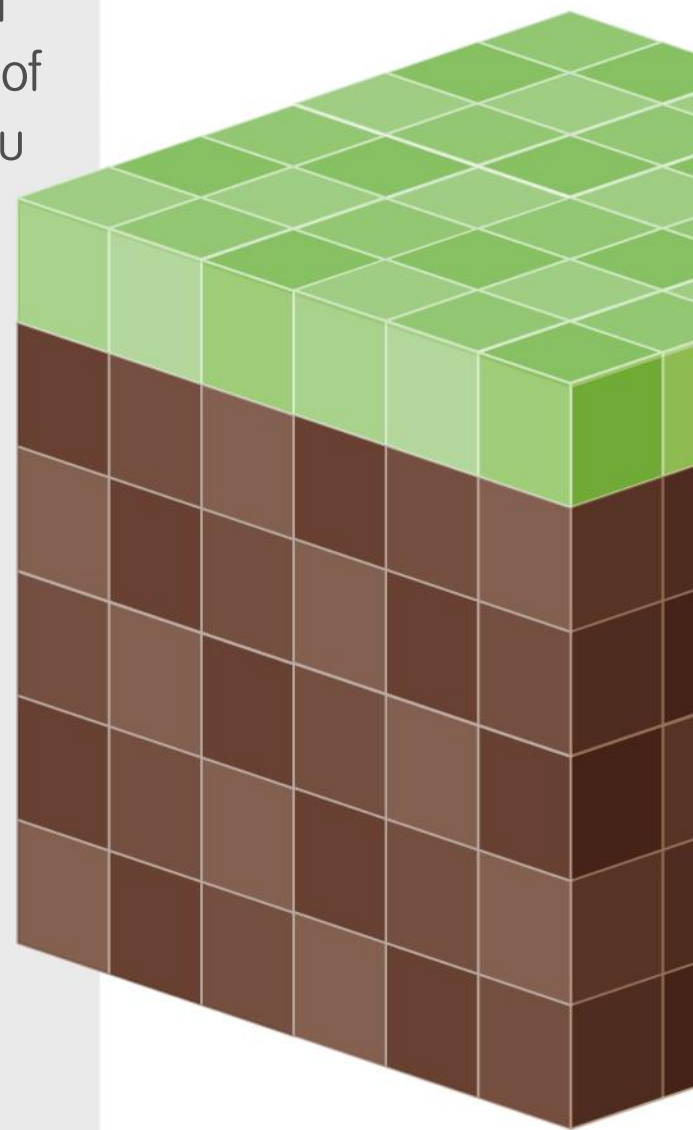
As well as living to play another turn, you also gain a point. This is achieved by incrementing the score variable by one each time you turn the block gold and return to the beginning of the loop to check the status of the next block you step on. The postToChat is to tell you that you have survived another move!

```
score = score + 1
mc.postToChat("You are safe")
```

17 Run the game

That completes the code for the game. The game is now ready to be run. To run the game, click the Run button in the top right corner of the IDE. The game will run in the browser window. The game is now ready to be played.

That completes the code for the program. Save it and then start a Minecraft game. Once the world has been created, run the Python program. Move back to the Minecraft window and you will see the board created in front of you. Watch out for that mine!

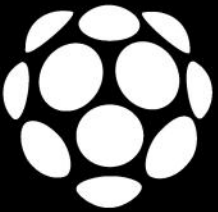




Underwater Pi drone

Niels Affourtit has turned his passion for RC vehicles into a fully functioning, submersible Ras Pi vehicle





Why did you decide to build an ROV (remotely operated vehicle)?

When I was a kid I frequently built RC models; sailing boats and speedboats. While I owned a fantastic RC glider and dreamed of having my own RC submarine someday, my neighbour at the time had an RC submarine and discouraged me from building one by saying this thing always had trouble with its electronics and leakage. Besides, almost all waters are murky in our area!

Around the end of 2014, I saw a National Geographic episode about the salvage of the Costa Concordia and was very impressed with how they used a VideoRay to monitor the work done by their divers, and it helped revive my dream of having my own model submarine. That's when I stumbled across the people who have built their own ROV.

How are the inner components of the ROV kept safe from water damage?

Since the Nineties, a lot has changed in the world of RC models. There are now LiPo batteries that give much more energy and power, and brushless outrunner motors that can run submerged in water, making moving parts through the hull unnecessary – this is the key development that makes the chance of leakage very small. The end closure is tightened by external water pressure and happens to be extremely reliable. A lot of the information and lessons I had learned are found on openrov.com, svseeker.com, raspberrypi.org and many other sites with forums about Linux and Raspberry Pi.

How long has the development of the ROV taken? Were there any problems?

When I started in December, I had a lot of stuff to learn and decided that purchasing a low-cost driver was where



Niels Affourtit

is an avid scuba diver who has turned his love for all things nautical into constructing an underwater ROV.

I should start. I opted for a Raspberry Pi instead of the BeagleBoard used by OpenRov, or an Arduino board. I had previously learned a little Linux from the workstation my company bought to do stress calculations (using our finite element program ANSYS). We also use Linux to improve daily interaction with third-party software, doing batch runs and fatigue calculations on steel pipelines. For all the functions on this ROV, there are so many options that I always face new challenges; this is what makes this project so interesting.



Raspberry Pi B+
Tilt, roll, pressure and temperature sensors
3D-printed camera tilt and propellers
Camera module
USB joystick control
Ethernet cable

What role does the Raspberry Pi take in this project?

The ROV is controlled with a joystick connected to a laptop. The laptop runs Python scripts, and with PyGame I can read the signals from the joystick. The signals are then translated into servo commands and sent to the Raspberry Pi via a simple socket connection. The Raspberry Pi is the brain of the ROV; it communicates with the surface laptop via Ethernet. Thanks to the OpenROV project, I learnt to implement a Tenda home plug, which reduces the communication lines from four to two wires, increases the reach from about 50



metres with a submerged CAT5 to 300 metres, and makes the signal much less susceptible to noise.

A battery-powered Wi-Fi router (bought for €1 at the local recycling store) then sends the signal to my laptop. The advantage of the Wi-Fi router is that I can attach it to a reel, avoiding the use of sliding contacts. It also reduces tripping risks on shore. The Wi-Fi router can be mounted on a buoy so that the ROV can be launched from a boat. Signals from the ROV are communicated via a web interface, the same way as OpenROV. All sensor data is written to a MySQL database and an Ajax script reads the data out and presents it on a web page. This web page also presents the live video feed from the camera of the ROV. Currently I'm using a Ras Pi cam, but I have ordered the USB camera used by OpenROV to improve the images.

The GPIO (General Purpose Input and Output) of the Pi is used to control the standard ESC (Electronic Speed Controller) of the four motors by PWM (Pulse Width Modulation), the tilt servo of the camera, relays for the lights, and server communication. Although the Pi now has many libraries for popular sensors, I still need an Arduino to communicate with



the depth sensor (MS5803-5). This Arduino is connected via the USB of the Pi and its analogue port is used to measure the voltage of the battery (using two resistors in series to drop the voltage to a safe 2-3V).

Since there is little space inside the ROV, the standard (USB) power connection on the side cannot be used. Therefore I inject power via the GPIO board at the 5V pin. This isn't a fused connection, imposing some risk, but it has worked perfectly for me.

Why did you decide to use 3D-printed parts in your ROV?

The propellers, tilt mechanism of the camera, and the Raspberry Pi housing are 3D printed. My neighbour imports Inno3D printers and I borrowed one to use for my project. This way I could vary the size, pitch and direction of the propellers. The parts take half an hour to print but cost only a few cents. Later on I will use the printer to make hydrodynamic motor mounts.

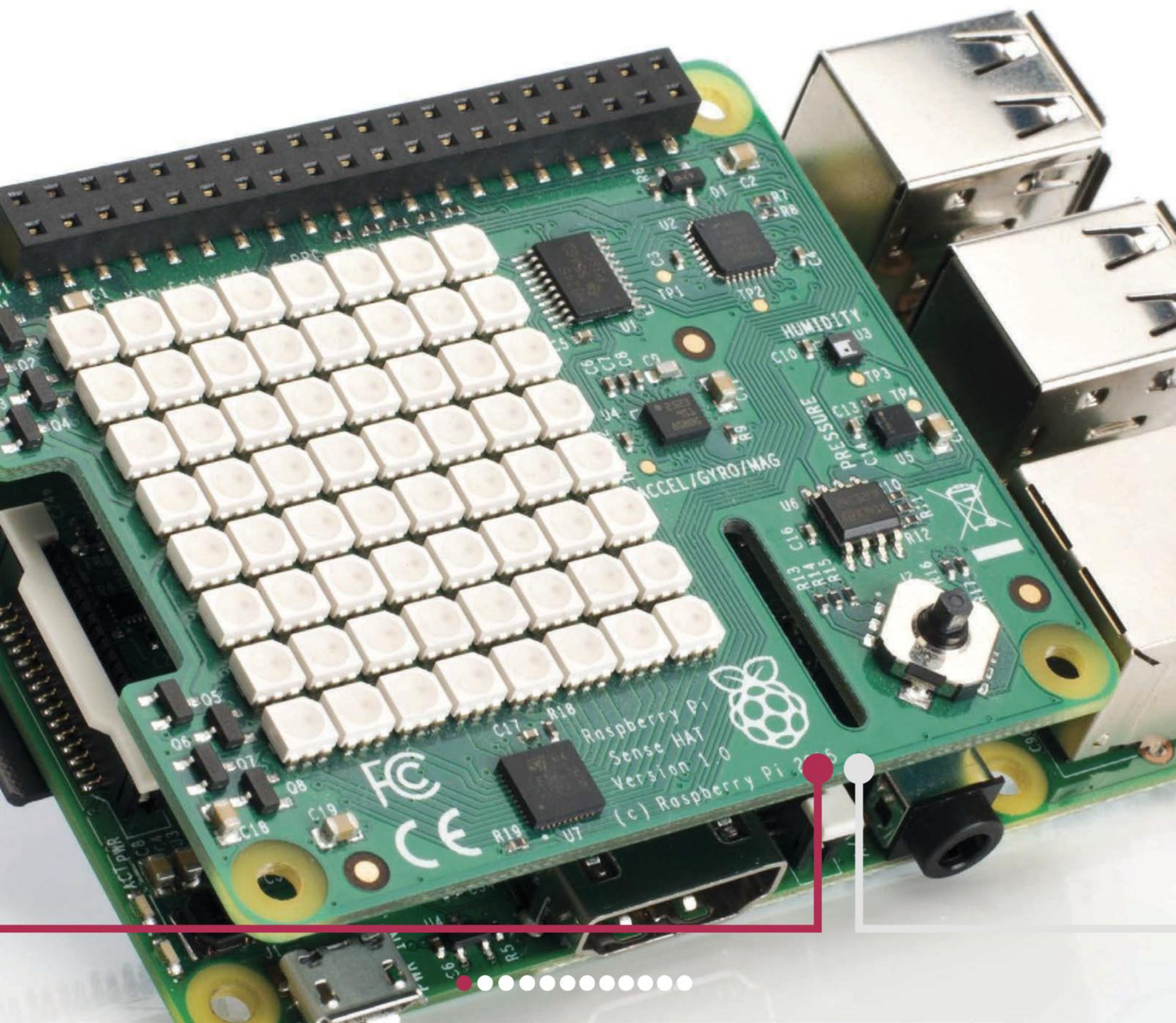
How deep can the ROV currently travel?

The tested depth is 13 metres. I took it down to a lake without the electronics and lowered it packed with lead. The collapse pressure determines the maximum depth of the ROV. I design offshore pipelines at INTECSEA and collapse pressure calculations are my everyday work. I made a simple finite element model of the ROV and applied external pressure to see how it would behave. Currently the 6mm-thick end caps are the bottleneck. At 30-40 metres they deflect so much that I anticipate they can start tearing. The lakes in the Netherlands are normally a few metres deep, at a few places 15-25 metres. Even the North Sea here is only 20-30 metres deep, so extending the depth limit has not been my priority.



Master essential Sense HAT skills

Develop your Sense HAT skills and start a coding
a version of Rock, Paper, Scissors, Lizard, Spock





In December 2015, two Raspberry Pi computers were each fitted with an Astro Pi, which is an add-on board boasting an array of sensors and an 8x8 LED matrix. These were flown to the ISS and delivered to Major Tim Peake with a selection of school children's experiments. In late December 2015, the Astro Pi was released for sale to the public as the rebranded 'Sense HAT'. This consists of exactly the same hardware and sensors set found on the Astro Pi but with a new 'Sense HAT' API. The first part of this two-part tutorial introduces you to the Sense HAT hardware and walks you through the skills required to create a Sense HAT version of the updated classic *Rock, Paper, Scissors, Lizard, Spock* game. These skills also stand alone and you can adapt them for use in your own projects.



**THE PROJECT
ESSENTIALS**

**Raspberry Pi
Sense HAT**

01 Install the Sense HAT software

First, attach the board to the GPIO header and install the Sense HAT software. This is pre-installed on the latest Raspbian images; for older images it can be downloaded – boot up your Raspberry Pi, load the LX Terminal and type in `sudo apt-get install sense-hat` to install the software. On completion reboot your Raspberry Pi.

```
sudo apt-get update
sudo apt-get install sense-hat
sudo reboot
```

02 Scrolling a message

Writing code to scroll text on LCD / LED displays can be challenging and frustrating. The Sense HAT API removes the difficulties and simplifies the whole procedure to a



reading is sound. To measure the temperature, return to your Python editor and type in the code below, then save and run the file. This will return the current temperature reading and print it out:

```
from sense_hat import SenseHat
sense = SenseHat()
temp = sense.get_temperature()
print("Temperature: %s C" % temp)
```

04 Compass reading

One of the more nifty sensors is the magnetometer, which can be used as a compass. This returns a measurement of the Sense HAT's position in relation to magnetic north. Again the code is easy to use: `sense.get_compass()` in line 3 returns the position, which is stored in a variable called `north`. The value that is measured is then printed out in line 4. Use the code example below to test the compass sensor and the readings:

```
from sense_hat import SenseHat
sense = SenseHat()
north = sense.get_compass()
print("North: %s" % north)
```

05 Mapping an LED image from a picture

Images are built up of pixels which combine to create an overall picture. Each LED on the matrix can be automatically set from an image file. For example, an image of a lizard can be loaded, the colours and positions calculated, and then the corresponding LEDs enabled. The image needs to be 8 x 8 pixels in size so



that it fits the LED Matrix. Download the supplied test picture file, lizard.png, and save it into the same folder as your program. Use the code below to open and load the image of the lizard (line 3). The Sense HAT will do the rest of the hard work for you:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.load_image("lizard.png")
```

06 Make your own 8 x 8 image

There are two further methods to create an image with the LEDs. The first is a superb on-screen program that enables you to manipulate the LEDs in real time. You can change the colours, rotate them and then export the image as code or as an 8 x 8 PNG file.

Install 'Python PNG library' by opening the terminal and typing `sudo pip3 install pypng`. After this has finished, type `git clone https://github.com/jrobinson-uk/RPi_8x8GridDraw`.

Once the installation has completed, move to the RPi folder with `cd RPi_8x8GridDraw`, then type `python3 sense_grid.py` to run the application.

07 Create and export your image

The Grid Editor enables you to select from a range of colours displayed down the right-hand side of the window. Simply choose the colour and then click the location of the LED on the grid, select 'Play on LEDs' to display the colour on the Sense HAT LED. Clear the LEDs using the 'Clear Grid' button and then start over. Finally, when exporting the image, you can either save as a

The origins of RPSLS

Sheldon Cooper from The Big Bang Theory famously brought the game into the mainstream media when he explained how it was played to his friend, though it wasn't actually invented on the show. The original expansion of the classic Rock, Paper, Scissors game is attributed to Sam Kass and Karen Bryla.



PNG file and apply the code in the previous step to display the picture, or you can export the layout as code and import that into your program.

08 Code each LED: Part 1

The second method to create an image is to individually code each LED and combine these. To set the colour of an LED, create a variable for a colour and assign an RGB value to it. Add additional colours by creating additional variables. Now create a representation of the image using the variable names: in this example, the X and O symbols combine to make a question mark. Set the LEDs with the code `sense.set_pixels(question_mark)`:

```
from sense_hat import SenseHat
sense = SenseHat()
X = [255, 0, 0] # Red
O = [255, 255, 255] # White
question_mark = [
0, 0, 0, X, X, 0, 0, 0,
0, 0, X, 0, 0, X, 0, 0,
0, 0, 0, 0, 0, X, 0, 0,
0, 0, 0, 0, X, 0, 0, 0,
0, 0, 0, X, 0, 0, 0, 0,
0, 0, 0, X, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, X, 0, 0, 0, 0
]
sense.set_pixels(question_mark)
```

09 Code each LED: Part 2

To add more colour variations to your image, create a

Python version of RPSLS

If you can't wait for the next part of this tutorial, then you can play around with this Python version of the RPSLS game. The Sense HAT version uses similar mechanics to calculate the winner and losers of each round played:

<https://trinket.io/python/46302fflaf>




```
B, B, 0, 0, 0, 0, X, X  
]  
sense.set_pixels(new_image)
```

10 Create an image of a lizard

Now you can begin to customise your own images to use. To get started, create a simple lizard using the LED layout. Then experiment with your own ideas and concepts. Remember, you will also need images for Rock, Paper, Scissors and Spock. Johan Vinet has some excellent and inspirational examples of 8 x 8 pixel art <http://johanvinet.tumblr.com/image/127476776680>.

11 Selecting a random image

Next you need a code to scroll through your images. Begin by renaming each of the images starting from one – for example 1.png, 2.png. Create a new variable called `playersChoice` which will store the current picture filename. Next, load the image using the code `sense.load_image(str(playersChoice) + ".png")` – note that `playersChoice` needs to be converted into a string before it is used. Once the picture is displayed, the variable is incremented, which loads the next picture. Finally, add a conditional to check if you have reached your last picture. For example, if you have five images then use:

```
if playersChoice == 5:  
    playersChoice = 0
```

... to check when the variable reaches the last image and to reset the variable to zero. This loads the first picture and the loop starts again from the beginning:

“The Sense HAT is equipped with a small multi-directional joystick that can be programmed to respond to direction”




```
import time
while True:
    sense.load_image(str(playersChoice) + ".png")
    playersChoice = playersChoice + 1
    time.sleep(1)
    if playersChoice == 5:
        playersChoice = 0
```

12 Clearing the image

When cycling through the images, the LEDs will turn on and off to the corresponding requirements of your code. However, there will be times when you need to turn off all the LEDs at once. This is referred to as clearing them. To clear the LEDs and set them all to 'off', use the line `sense.clear()`.

13 Setting the joystick up in PyGame

The Sense HAT is equipped with a small multi-directional joystick that can be programmed to respond to direction. The RPSLS game makes use of the joystick to cycle through your images and to select one. In a new Python file, set up the PyGame window by importing the PyGame modules at the top of your file (lines 1 and 2). Next, initialise with `pygame.init()`. PyGame runs in a separate window, but since the game takes place on the Sense HAT, the window is obsolete. You can therefore minimise the size by using: `pygame.display.set_mode((140, 180))` line 5.

```
import pygame
from pygame.locals import *

###Set up PyGame Screen###
```




```
for event in pygame.event.get():  
  
    if event.type == KEYDOWN and  
playersChoice < 5:  
        if event.key == K_UP:  
print("UP")
```

15 Selecting a picture

You have now coded PyGame to wait for 'events' and check and respond if the joystick is moved 'up'. In the RPSLS game, this procedure is used to enable the player to select a picture that represents their turn – for example, Paper. Replace the `print("UP")` on line 7 with the image section code used in step 11. As before, use `sense.load_image(str(playersChoice) + ".png")` to load the image onto the LEDs, then increment the `playersChoice` variable using `playersChoice = playersChoice + 1` to select the next picture. Line 6 checks if you have reached the end of the picture cycle and then resets it back to a value of zero (line 8), so the cycle begins again starting from picture one:

```
if event.type == KEYDOWN and playersChoice <  
5:  
    if event.key == K_UP:  
        print (playersChoice)  
        sense.load_image(str(playersChoice) +  
".png")  
        playersChoice = playersChoice + 1  
    if playersChoice == 5:  
        playersChoice = 0
```



16 Confirming your selection

Once you have scrolled through and chosen your picture then you need to be able to select it. This is achieved pressing the whole joystick button down, which acts as if the Return key has been pressed. On line 2, the code `if event.key == K_RETURN`: checks for this action and then responds by breaking the loop. The loop stops cycling through and the current image number is then stored inside the `playersChoice` variable, where it is used later in the game to compare with the Computer's choice and calculate the winner.

```

    """Checks for a 'select / Enter' Choice"""
    if event.type == KEYDOWN:
        if event.key == K_RETURN:
            running = False
            break
        """Ends loop and moves onto main
game"""

```

17 RPSLS

You now have a basic structure for the start of the RPSLS game. The program enables you to scroll through up to five images, which each represent one of the hands that you can play: Rock, Paper, Scissor, Lizard or Spock. In the next issue, part two of this tutorial will code the mechanics of the gameplay and combine this to create the finished RPSLS game. A sneak video of the final game in action can be viewed here: **www.youtube.com/watch?v=T_ZvWkMgVFM**. In the meantime, work on creating your 8 x 8 pixel art.





Anonymise your web traffic with a Pi Tor router

Use your Raspberry Pi as a wireless access point and route traffic anonymously



Are you concerned about your online privacy? Suspect that the NSA or GCHQ are tracking you online? Or are you simply anxious about advertising networks targeting their ads at you? If this is the case, or you're simply on the lookout for a new project, then setting your Raspberry Pi up as an anonymising Tor wireless access point is a good way to save time and effort setting up Tor on all of your PCs and mobile devices.

If you're unaware, Tor is free software that enables anonymity online and routes internet traffic through a worldwide anonymising network of over 7000 relays. By doing this, your location and website use can be concealed from techniques such as traffic analysis and network surveillance.

This is all about protecting your privacy by making your visits to websites, instant messages, emails and more difficult to trace, and can be set up on your Raspberry Pi with a wireless dongle (or a Raspberry Pi 3), Ethernet cable, and a bit of configuration.



THE PROJECT ESSENTIALS

Ethernet cable

Wireless router

USB WiFi adaptor or
Raspberry Pi 3


```
pi@raspberrypi: ~  
atomicckarma@atomic-laptop-ubuntu:~$ ssh pi@192.168.0.27  
The authenticity of host '192.168.0.27 (192.168.0.27)' can't be established.  
ECDSA key fingerprint is e7:50:4b:9d:d8:58:36:75:99:1a:2c:42:7d:c3:8c:20.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '192.168.0.27' (ECDSA) to the list of known hosts.  
pi@192.168.0.27's password:  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Mon Dec 7 16:17:07 2015  
pi@raspberrypi ~ $
```

Pi, with a mouse and keyboard.) To use SSH, open a terminal and enter:

```
ssh pi@192.168.0.27
```

... where the Raspberry Pi has the default username of "pi" and the IP address 192.168.0.27. If your distro doesn't have SSH installed, you can install openssh-client or putty. You can also communicate via SSH in Windows using PuTTY.

You'll find the IP address of your Raspberry Pi by checking your router's admin page, or connecting a monitor and keyboard and entering `ifconfig`.

03 Install access point

Next, connect to the Raspberry Pi with SSH and enter `iwconfig` to ensure the wireless adaptor is recognised. Then, refresh the package list with `sudo apt-get update` and install the wireless access point software:

```
sudo apt-get install hostapd isc-dhcp-server
```

04 Edit DHCP settings

Next, we need to edit the `/etc/dhcp/dhcpd.conf` file, to

```
# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.
```

^{^G} Get Help ^{^O} WriteOut ^{^R} Read File ^{^Y} Prev Page ^{^K} Cut Text ^{^C} Cur Pos
^{^X} Exit ^{^J} Justify ^{^W} Where Is ^{^V} Next Page ^{^U} UnCut Text ^{^T} To Spell

avoid IP addresses being assigned randomly. Open in nano with:

```
sudo nano /etc/dhcp/dhcpd.conf
```

Scroll through for the lines starting "option domain-name" and comment them out with a # as follows:

```
#option domain-name-servers ns1.example.
org, ns2.example.org;
#option domain-name "internal.example.
org";
```

```
# range 10.17.224.10 10.17.224.250;
# }
# pool {
#     deny members of "foo";
#     range 10.0.29.10 10.0.29.230;
# }
#}
subnet 192.168.12.0 netmask 255.255.255.0 {
range 192.168.12.5 192.168.12.50;
option broadcast-address 192.168.12.255;
option routers 192.168.12.1;
default-lease-time 600;
max-lease-time 7200;
option domain-name "local";
option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

^{^G} Get Help ^{^O} WriteOut ^{^R} Read File ^{^Y} Prev Page ^{^K} Cut Text ^{^C} Cur Pos
^{^X} Exit ^{^J} Justify ^{^W} Where Is ^{^V} Next Page ^{^U} UnCut Text ^{^T} To Spell

Following this, find “authoritative” and uncomment the line by removing the #.

05 Specify IP addresses

Head down to the end of the document and add the following lines to define the IP address for the access point, the range of addresses and the domain name servers (also known as DNS):

```
subnet 192.168.12.0 netmask 255.255.255.0 {  
    range 192.168.12.5 192.168.12.50;  
    option broadcast-address 192.168.12.255;  
    option routers 192.168.12.1;  
    default-lease-time 600;  
    max-lease-time 7200;  
    option domain-name “local”;  
    option domain-name-servers 8.8.8.8,  
8.8.4.4;  
}
```

Note that the domain name servers specified here are provided by Google. Press Ctrl+X to save and exit, following the prompts.

06 Assign a static IP

Because the Raspberry Pi’s wireless card receives a dynamic IP address by default from your router, we need to specify a static IP so that it can always be accessed in its role as a Tor anonymiser.

With nano, open /etc/default/isc-dhcp-server, scroll down to the line that currently reads INTERFACES=“”, and edit so it reads INTERFACES=“wlan0”. Save and exit.

Deal with errors

Various errors can occur during the setting up of this project. Problems with your Locale settings are one example, and this can be dealt with by editing the ssh_config file. Enter:

```
sudo nano /  
etc/ssh/ssh_  
config
```

Find the line that reads “SendEnv LANG LC_*”, commenting it out with a # at the beginning. You may need to reboot your Raspberry Pi to fully overcome the error.



```
# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

07 Set up the wireless adaptor

With wireless adaptor wlan0 now specified, we need to disable it before configuring, using the command `sudo ifdown wlan0`. Using nano once again, open the `/etc/network/interfaces` text file and comment out three lines with `#`, so they read:

```
# iface wlan0 inet manual
# wpa-roam /etc/wpa_supplicant/wpa_
supplicant.conf
# iface default inet dhcp
```

Find the line `allow-hotplug wlan0` and add the following:

```
iface wlan0 inet static
address 192.168.12.1
netmask 255.255.255.0
```

Use `Ctrl+X` to save, and activate:




```
sudo ifconfig wlan0 192.168.12.1
```

08 Configure your new WAP

With the wireless access point physically built, the software installed and then defined, it is time to configure it. In nano, create a new file:

```
sudo nano /etc/hostapd/hostapd.conf
```

To this, add the following:

```
interface=wlan0
ssid=TorHotSpot
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=$Your_Passphrase$
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

You can specify your own SSID; you'll also need to specify your own secure password.

09 Find the configuration file

By default, your Raspberry Pi will not be able to see the new configuration file. In order for it to be detected when the wireless access point boots, we need to point to it in the /etc/default/hostapd file. To do this,

Which Pi can you use?

While the older, dual USB port Raspberry Pis can be used for this project, the best results are achieved with the Raspberry Pi 2 and 3 models. The reasons are simple: both have four USB ports, thereby enabling the connection of other devices such as keyboard and mouse if necessary, and both are fast enough to deal with the volumes of data routing required. A Pi Zero probably wouldn't be practical as a long-term solution.




```
ip_forward”
```

11 Specify routing rules

With this done, routing rules need to be added. These will connect the Ethernet port `eth0` and the Raspberry Pi wireless port `wlan0`:

```
sudo iptables -t nat -A POSTROUTING -o  
eth0 -j MASQUERADE  
sudo iptables -A FORWARD -i eth0 -o  
wlan0 -m state --state RELATED,ESTABLISHED  
-j ACCEPT  
sudo iptables -A FORWARD -i wlan0 -o  
eth0 -j ACCEPT
```

12 Use the rules upon boot

Routing rules will be deleted when you restart the Pi, so in the terminal enter:

```
sudo sh -c “iptables-save > /etc/  
iptables.ipv4.nat”
```

This ensures the rules will persist. Next, open `/etc/network/interfaces` with `nano` and add this line to the end to load rules when the device boots:

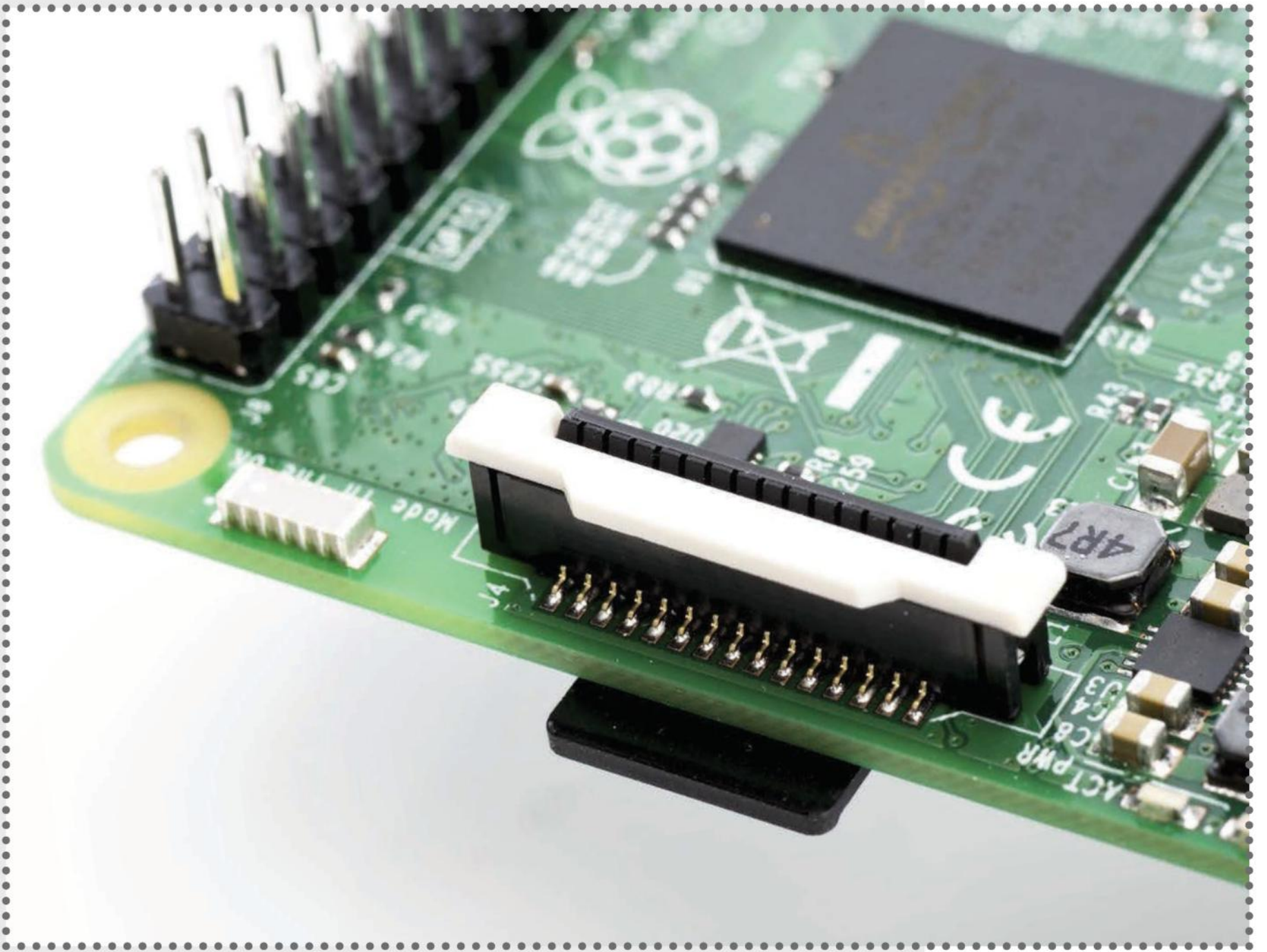
```
up iptables-restore < /etc/iptables.ipv4.  
nat
```

Save and exit, then restart the DHCP server:

```
sudo service isc-dhcp-server restart
```

“NAT needs to be configured to allow devices to connect to the Raspberry Pi’s wireless access point”





Finally, enable the access point:

```
sudo /usr/sbin/hostapd /etc/hostapd/  
hostapd.conf
```

13 Ready your access point

At this stage, it should be possible to connect to the Raspberry Pi wireless access point. But you'll need to guarantee some settings, starting with the DHCP and hostapd services:

```
sudo service hostapd start
sudo service isc-dhcp-server start
```

Above You could even use the Pi 3's Bluetooth to control your smart home gadgets as well as route your web traffic through Tor

Update the initialisation (init) scripts with:

```
sudo update-rc.d hostapd enable  
sudo update-rc.d isc-dhcp-server  
enable
```

Restart with `sudo shutdown -r now`, and retry the connection from your mobile device.

14 Time to install Tor

As things stand, the Raspberry Pi is a wireless access point. But it is not anonymised yet. Reconnect via SSH and then run `sudo apt-get install tor`, then open `/etc/tor/torrc` in nano and add the following to instruct Tor to anonymise the wireless access point:

```
Log notice file /var/log/tor/notices.  
log  
VirtualAddrNetwork 10.192.0.0/10  
AutomapHostsSuffixes .onion,.exit  
AutomapHostsOnResolve 1  
TransPort 9040  
TransListenAddress 192.168.12.1  
DNSPort 53  
DNSListenAddress 192.168.12.1
```

15 Add SSH exception

Next, flush the IP tables with:

```
sudo iptables -F
```

“Tor is free software that enables anonymity online and routes internet traffic through a worldwide anonymising network”



```
sudo iptables -t nat -F
```

And follow this by adding an exception for SSH connections on port 22:

```
sudo iptables -t nat -A PREROUTING  
-i wlan0 -p tcp --dport 22 -j REDIRECT  
--to-ports 22
```

These two rules enable DNS lookups and direct TCP traffic to Tor's port 9040, respectively.

```
sudo iptables -t nat -A PREROUTING  
-i wlan0 -p udp --dport 53 -j REDIRECT  
--to-ports 53
```

```
sudo iptables -t nat -A PREROUTING  
-i wlan0 -p tcp --syn -j REDIRECT --to-  
ports 9040
```

Save to the NAT file with:

```
sudo sh -c "iptables-save > /etc/  
iptables.ipv4.nat"
```

16 Enable the Tor service

At this stage, you can enable the Tor service on your wireless access point with:

```
sudo service tor start
```

Boot scripts can be enabled with:

```
sudo update-rc.d tor enable
```

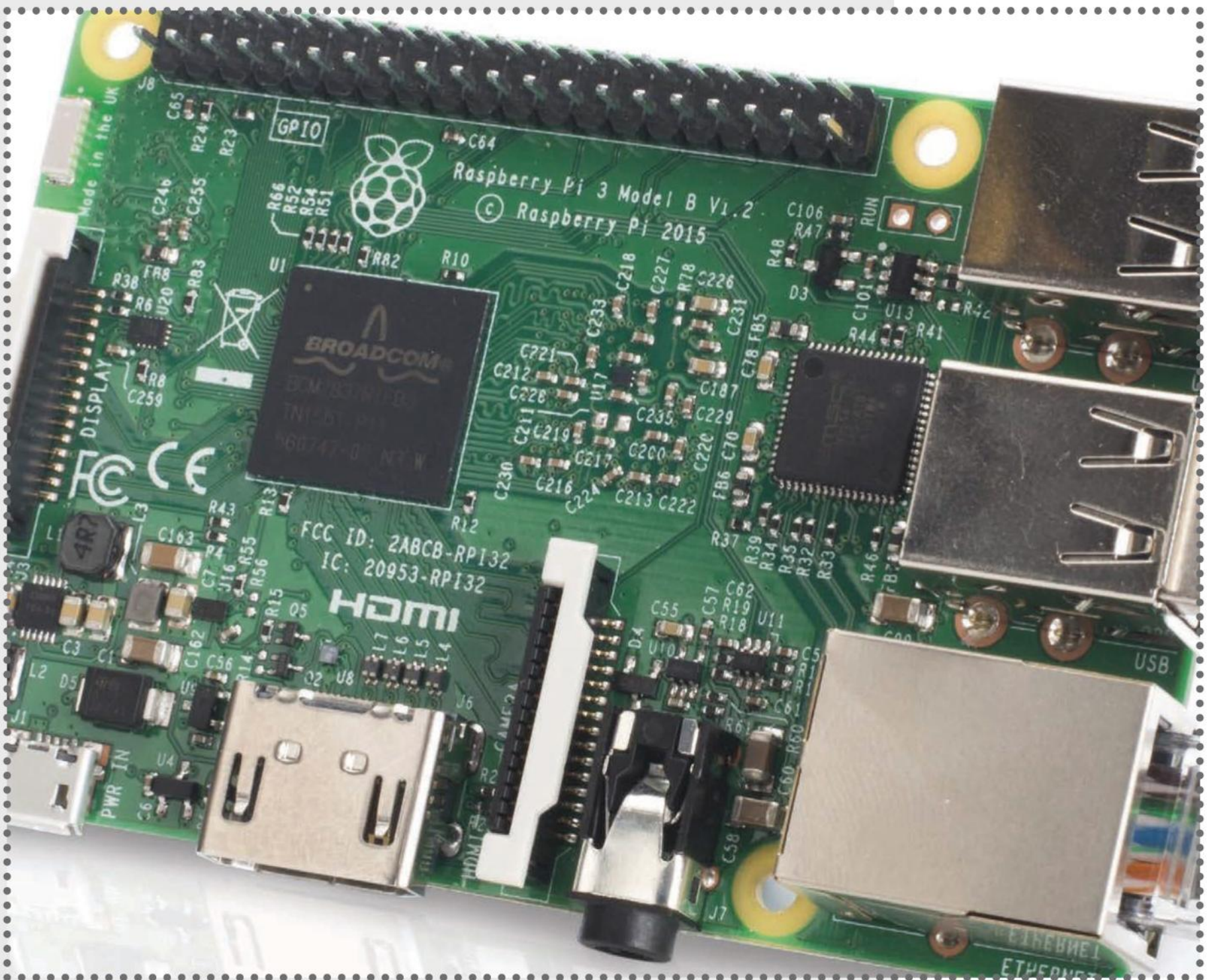


Restart the Pi again with `sudo shutdown -r now`.

17 Verify Tor anonymity

With your Raspberry Pi wireless access point restarted, it's time to connect using a Wi-Fi enabled device, perhaps a smartphone or laptop. As before, the connection should work without any problems – but are you browsing anonymously?

Find out by heading to **<https://torproject.org>** from two devices and comparing the IP addresses. If they're different, you're anonymous!





Create your own circuit diagrams with Fritzing

Wondering how we illustrate circuit diagrams? With Fritzing – a donationware utility you can use too!



Putting circuits together can be tricky, even for experts. Fortunately, various notations exist that make it easier to work out just where wires, components and power sources are connected. But rather than jot these details down with a pencil and paper, you might prefer to bring an element of realism to the illustration. This is particularly useful for beginners, and if you're demonstrating a breadboard-mounted circuit with an Arduino, for instance, you might wish to clearly and accurately illustrate the precise positions of each component and wire. This is where Fritzing comes in.

You've probably seen illustrations of circuits in the pages of RasPi over the years, and with Fritzing you too can put together circuit illustrations, simply by dragging and dropping the categorised components into place, saving, and exporting. In addition to breadboard illustrations, you can also create traditional schematics and PCB diagrams.

With some of the same familiar shortcuts and tools that you may have already used in an art package, Fritzing is remarkably easy to use.

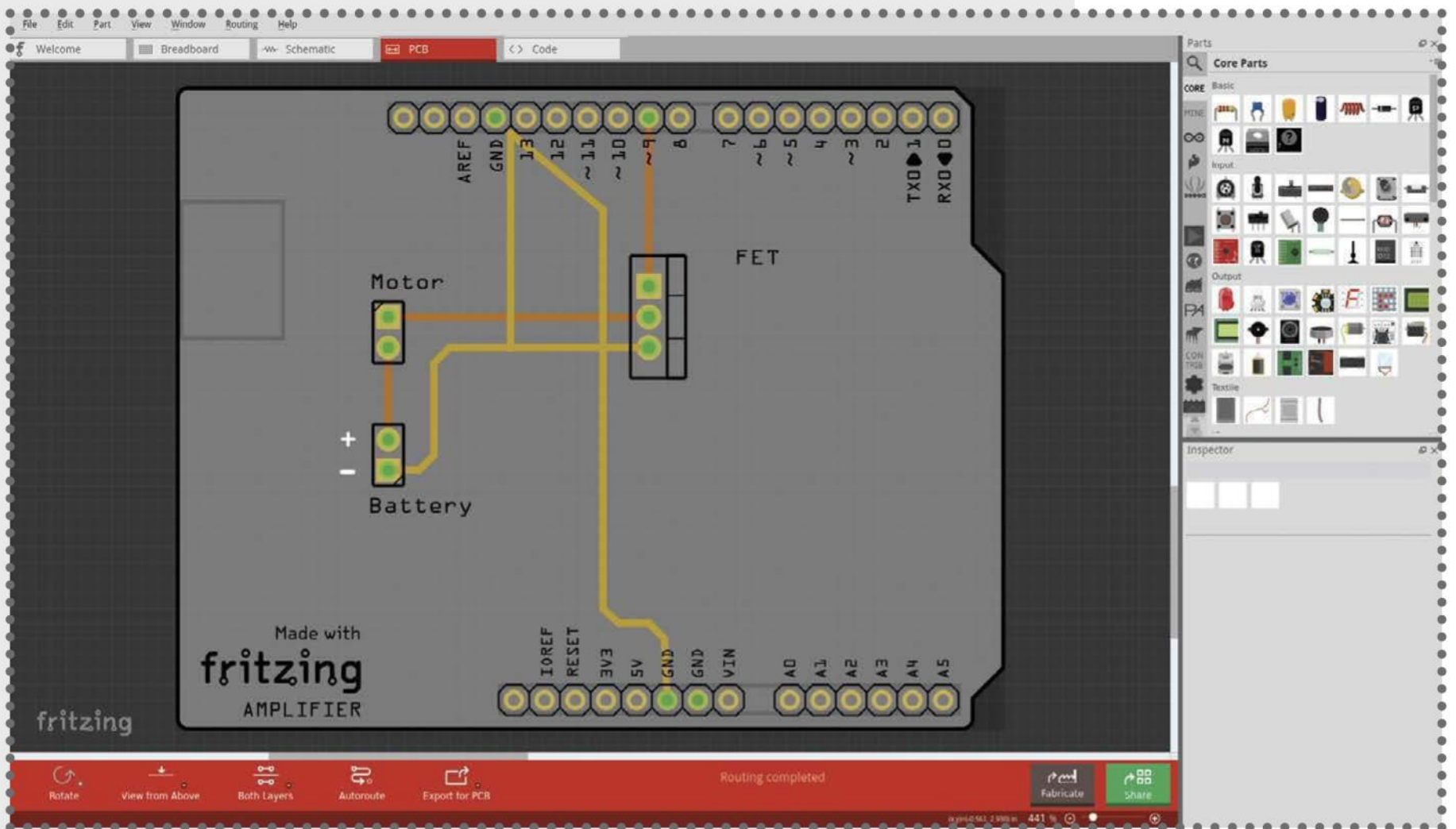


THE PROJECT ESSENTIALS

Ethernet cable

Wireless router

USB WiFi adaptor or
Raspberry Pi 3



01 Install Fritzing

For the quickest installation option, use `sudo apt-get install fritzing` on Ubuntu (or whichever Debian fork you're using). Alternatively, for Fedora use `yum install fritzing`. If you prefer, you can also download from the website (ensuring you select the 32-bit or 64-bit as appropriate for your system) and extract the contents. Once you've done this, navigate to the directory in the Terminal and run `./Fritzing` to launch.

02 Select your project type

With the app launched, you'll need to select an appropriate project type. Three are available in Fritzing: Breadboard, Schematic and PCB. We're starting with the Breadboard project, so select this. Across the bottom of the screen you'll see controls to add notes and rotate the board; on the right, various component illustrations are listed by category. You'll

The screenshot displays the Tinkercad software interface. On the left, a breadboard is shown with a grid overlay. The grid has columns labeled 30, 35, 40, 45, 50, and 55. A red horizontal line is drawn across the breadboard. On the right, the 'Parts' palette is open, showing a search bar and a list of components categorized under 'Core Parts'. The categories include 'Basic', 'Input', 'Output', and 'Textile'. The 'Basic' category is currently selected, showing various electronic components like resistors, capacitors, and integrated circuits. The 'Input' category shows components like buttons, potentiometers, and sensors. The 'Output' category shows components like LEDs, buzzers, and motors. The 'Textile' category shows components like fabric and thread.

03 Select your parts

04 Manipulate your board

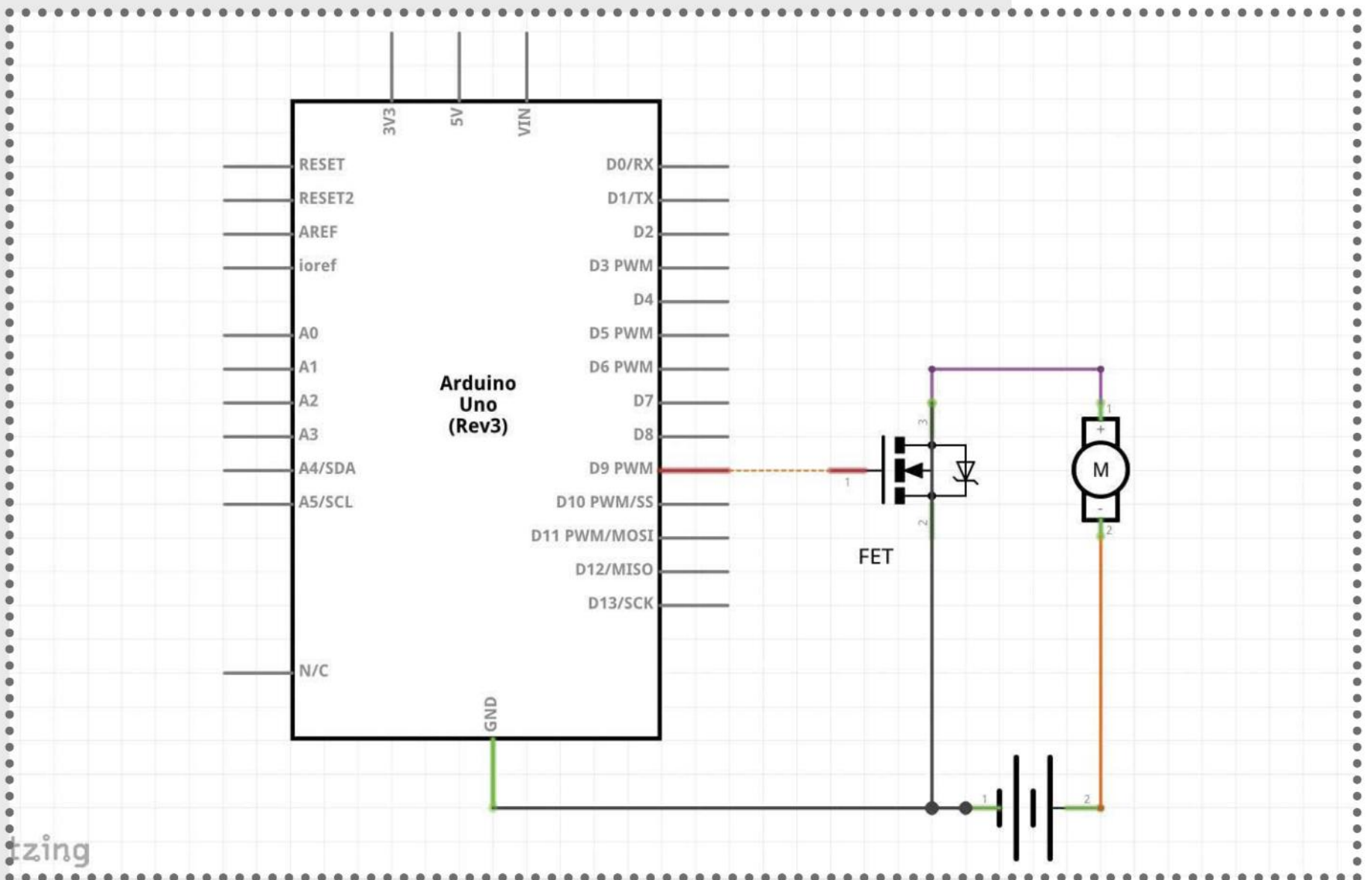
QUESTION

Breadboarding is an easy way of putting circuits together. But the graphical representation isn't to everyone's liking. Fortunately, Fritzing includes various views that provide alternatives. Across the top of the screen you will notice the Schematic and PCB buttons. Click these to switch views – the project you're currently working on will be displayed in a different format! This can prove useful later on at the export stage.

So far we've been playing around with the breadboard on its own. But what if you were developing a project with an Arduino and a breadboard-based circuit?

Get organised

QUESTION



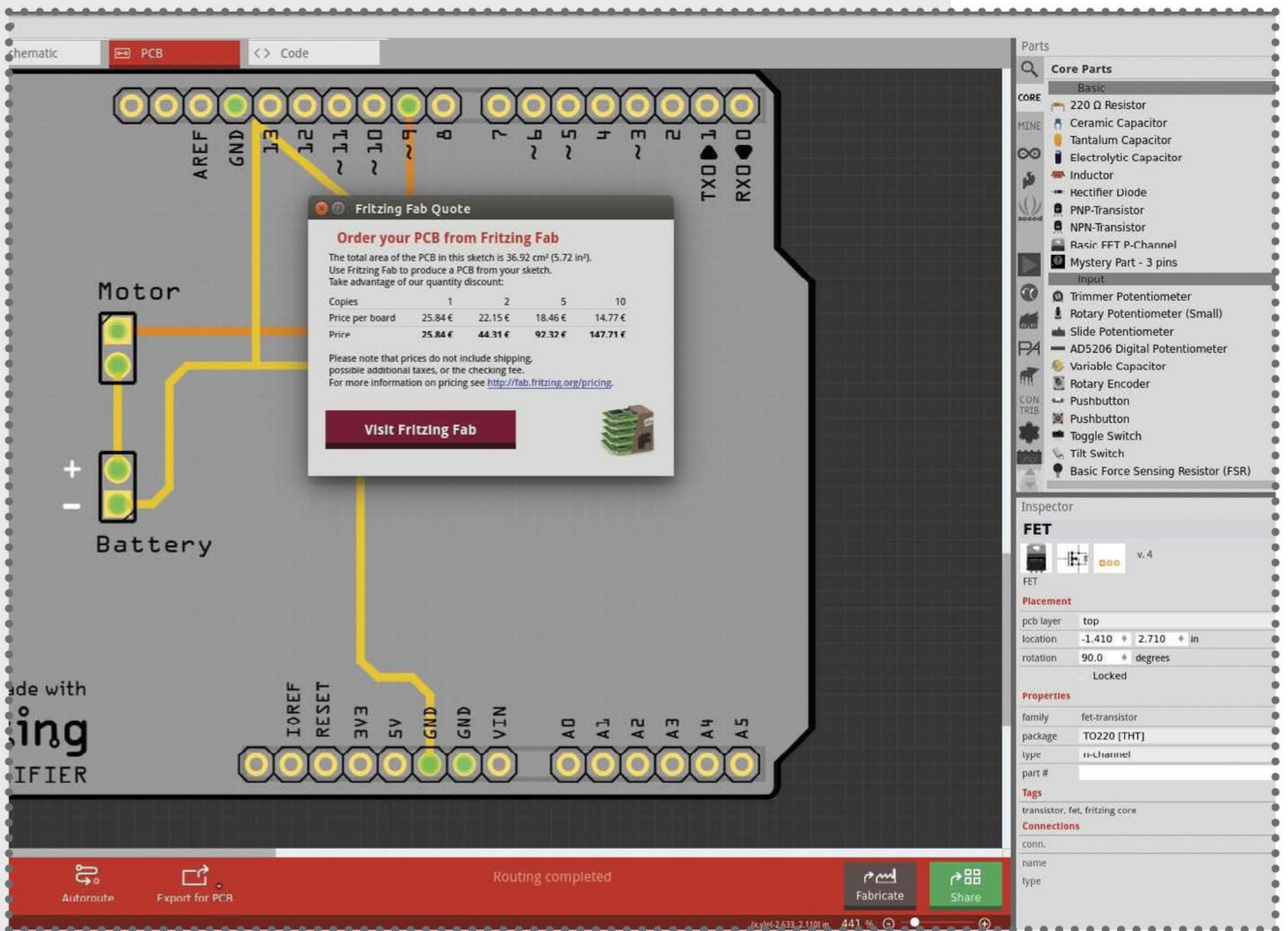
rectangle representing an Arduino can be moved around with a simple left-click and drag, the circuit originally built on a breadboard cannot. Instead, it is merely a collection of components that are connected with wires.

Above You can convert your diagrams into proper schematics, too

11 Playing with the PCB view

Similarly, the PCB view offers both a different visual experience, and as such how you move things around also changes. Rather than the Arduino and breadboard circuit being presented as separate entities, in this view the circuit sits atop of the Arduino.

Adjusting the position of components is trickier here, but this presents a more intimate look at just how the parts on the breadboard are connected to the Arduino.



12 Use the grid

Keeping your project tidy is important, more so your Fritzing diagram. To help here, enable the View > Align to Grid and Show Grid options, which will assist with repositioning your boards. How far the selected items move will depend on which of the three views you are currently using, and your zoom level. Repositioning is via the mouse, and fine-tuning using the arrow keys.

13 Autoroute assistance

While it isn't possible to automate the creation of your Fritzing project, the next best thing is

Above If you've designed a PCB – perhaps a new Pi HAT – you can order a prototype from inside the software!

board, and a quote for 1, 2, 5 or 10 boards to be printed, displayed in Euros. What this means is that a prototype board can be designed in Fritzing and a physical copy easily ordered!

17 Export your Fritzing project

With your Fritzing project complete, you'll probably want to share it with others. Sharing is possible with the Export tool, found in the File menu, where two main options are available: As Image and For Production. If you're planning to print your own circuit, use the latter option.

For sharing a JPG, PNG, SVG or PDF of the diagram, however, select As Image and the preferred format. It's really worth checking out some existing projects, as there's so much you can do with Fritzing that you may not have even thought of.

Detailed placement

Should you have a need to place components or change the colour of wires, you can employ the Inspector panel. With a component selected, use the up and down triangles in the Placement section of the Inspector panel to adjust its x and y position. Meanwhile, wires can be recoloured by left-clicking to select and then changing the option in the colour box, under Properties.



Motion tracking with your Pi

This issue, you will learn how to track motions with your Raspberry Pi, a camera and some Python code



In a previous article, we looked at how you can capture images using a camera and a Raspberry Pi. This let you include image capture functionality within your own Python program, but there is so much more you can do once you add vision to your code. This month, we will look at how you can add motion detection to your Python program.

This kind of advanced image processing is extremely difficult to do, so we will definitely be building on the hard work of others. Specifically, we will be using the excellent OpenCV Python package. This package is constantly being improved, with more functionality being added with every update.

The first thing you will need to do is install the various Python packages that you will need to talk to the camera and use OpenCV. Installing the packages can be done with:

```
sudo apt-get install python-picamera  
python-opencv
```

This will also install all of the required dependencies. This project will assume that you will use the camera module for the Raspberry Pi. Check out the boxout to the right for other options if you want to try using a USB webcam. To talk to the camera module, you need to import the PiCamera class from the picamera Python module. You will also need the PiRGBArray class so that you can store the raw data from the camera. To talk to the camera, you instantiate a new instance of the PiCamera class. You can then set the resolution and frame rate before you start capturing images.

```
from picamera import PiCamera
from picamera import PiRGBArray
camera = PiCamera()
camera.resolution = tuple([640,480])
camera.framerate = 16
rawImage = PiRGBArray(camera,
tuple([640,480]))
```

You now have your camera ready, and a memory buffer available to store the captured images in. There are several different methods that you can use to do motion tracking. One of the simpler ones is to try and notice when something within the image field changes. There is a Python module, called imutils, that provides several basic image processing functions that are useful in the pre-processing steps. There is no package for it within Raspbian, however, so you will want to install it with:

```
sudo pip install imutils
```

To look at image changes, we need to see what the background image looks like. You can take a series of images and look at the average of them to get an idea of the general background. Then, if a new image differs from the averaged background, we know that something has changed. This change is most probably due to something moving within the field of the image. To simplify the process, we will greyscale the image and then blur it slightly to get rid of any high-contrast regions. You will then want to simply run a continuous loop, pulling an image from the camera and running this process:

```
import imutils
import cv2
for f in camera.capture_
continuous(rawImage, format='bgr',
            use_video_port=True):
    frame = imutils.resize(f.array,
                           width=500)
    gray = cv2.cvtColor(frame, cv2.
                        COLOR_BGR2GRAY)
    gray = cv2.GaussianBlur(gray, (21,
                                   21), 0)
```

Here we start using the OpenCV functions to handle the image processing steps. You may have noticed that we are actually working with the array representation of the raw image data for the captured frame. There is no meta-data wrapping this image information, so it is your responsibility

to remember what you are working with. The next step within the loop is to check whether we have an averaged image yet, and to initialise it if we don't. So the first time through the loop, the following code will execute:

```
if avg is None:
    avg = gray.copy().astype("float")
    rawImage.truncate(0)
    continue
```

Now that we have an averaged image, we can add every subsequent captured image to the weighted average. We also need to find how different the current image is from this weighted average.

```
cv2.accumulateWeighted(gray, avg, 0.5)
imgDiff = cv2.absdiff(gray, cv2.
convertScaleAbs(avg))
```

By using this weighted average, we should be able to deal with false positive hits due to environment changes like fluctuations in the lighting.

Now that you have what is different from the average, what can you do with it? How do you decide how different it is from the average? We need to set some threshold difference that signifies a "real" difference in the image from the average. If you then dilate this thresholded image, you can apply the findContours function to identify the contours of the objects that are different from the calculated averaged background:

“We will be using the OpenCV Python package, which is constantly being improved”

```
imgThresh = cv2.threshold(imgDiff, 5,  
255, cv2.THRESH_BINARY)[1]  
imgThresh = cv2.dilate(imgThresh,  
None, iterations=2)  
(conts, _) = cv2.findContours  
(imgThresh.copy(), cv2.RETR_EXTERNAL,  
cv2.CHAIN_APPROX_SIMPLE)
```

This dumps all of the contours from the current image into the list 'conts'. You probably aren't very interested in tiny objects within the list of contours. These might simply be artifacts within the image data. You should loop through each of these and ignore any that are below some area limit. You probably want to highlight any remaining object contours by placing a bounding box around them. Luckily, OpenCV provides a function that will give the corner coordinates and the width and height. You can then draw a box on the image using this information:

```
for c in conts:  
    if cv2.contourArea(c) < 5000:  
        continue  
    (x, y, w, h) = cv2.boundingRect(c)  
    cv2.rectangle(frame, (x, y), (x+w,  
y+h), (0, 255, 0), 2)
```

You should now have an image with all of the moving objects highlighted by red bounding boxes.

What can you do with these annotated images, though? If you have a graphical environment available, you can display these results directly on the screen. OpenCV includes several functions to display

the results of your image analysis. The simplest is to use the `imshow()`, which will pop up a window to display the image and also add a title.

```
cv2.imshow("Motion detected", frame)
```

If you aren't monitoring the results of your motion detector in real time, you probably still want to capture images when something moves in the environment. Luckily, OpenCV also includes a pretty exhaustive list of IO functions. You will probably want to timestamp these images first, though. Using the Python module `timestamp` and the function `putText()`, you can get the current time and date and add it to the image itself with:

```
import timestamp
ts = timestamp.strftime("%A %d %B %Y
%i:%M:%S%p")
cv2.putText(frame, ts, (10, frame.shape[0]
- 10), cv2.FONT_HERSHEY_SIMPLEX, 0.35, (0,
0, 255), 1)
```

Now you have an image with the current time and date on it, and the parts of the image that show up as having movement bounded in red boxes. You can use the OpenCV IO functions to write out these images so that you can check them out later. The following code is an example:

```
cv2.imwrite("filename.jpg", frame)
```



The function `imwrite()` uses the file name extension in order to figure out what format to use when writing out the image. It can handle JPEG, PNG, PBM, PGM, PPM and TIFF. If the particular format you want to use also takes options, you can include them in the call to `imwrite()` as well. For example, you can set the JPEG quality by including `CV_IMWRITE_JPEG_QUALITY` and then setting it to some value between 0 and 100.

Everything we have looked at has been focused on the idea of analysing the images in real time, and this is great if you can put the Raspberry Pi in the same location as the camera. If you can't fit it in, though, you can still use the ideas here to post-process the video recorded by your micro-camera. You can use the same OpenCV IO functions to load the video file with:

```
camera = cv2.VideoCapture("filename.avi")
```

You can then run through the same process to analyse each of the image frames within the video file. The `VideoCapture()` function can also read in a series of image files if your camera is simply grabbing a series of still images rather than a video.

Once your program finishes, you need to remember to clean up after yourself. You should release the camera that you were using, and if you had OpenCV display any images on the desktop then you should clean those up, too.

```
camera.release()  
cv2.destroyAllWindows()
```



You should have enough information now to be able to add some basic motion detection to your own Python programs. If you explore the OpenCV documentation, you will find many other, more complex, image processing and analysing tools that are available to play with. Also, more functionality is constantly being added to the OpenCV project.



Talking Pi

Join the conversation at...



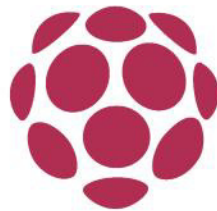
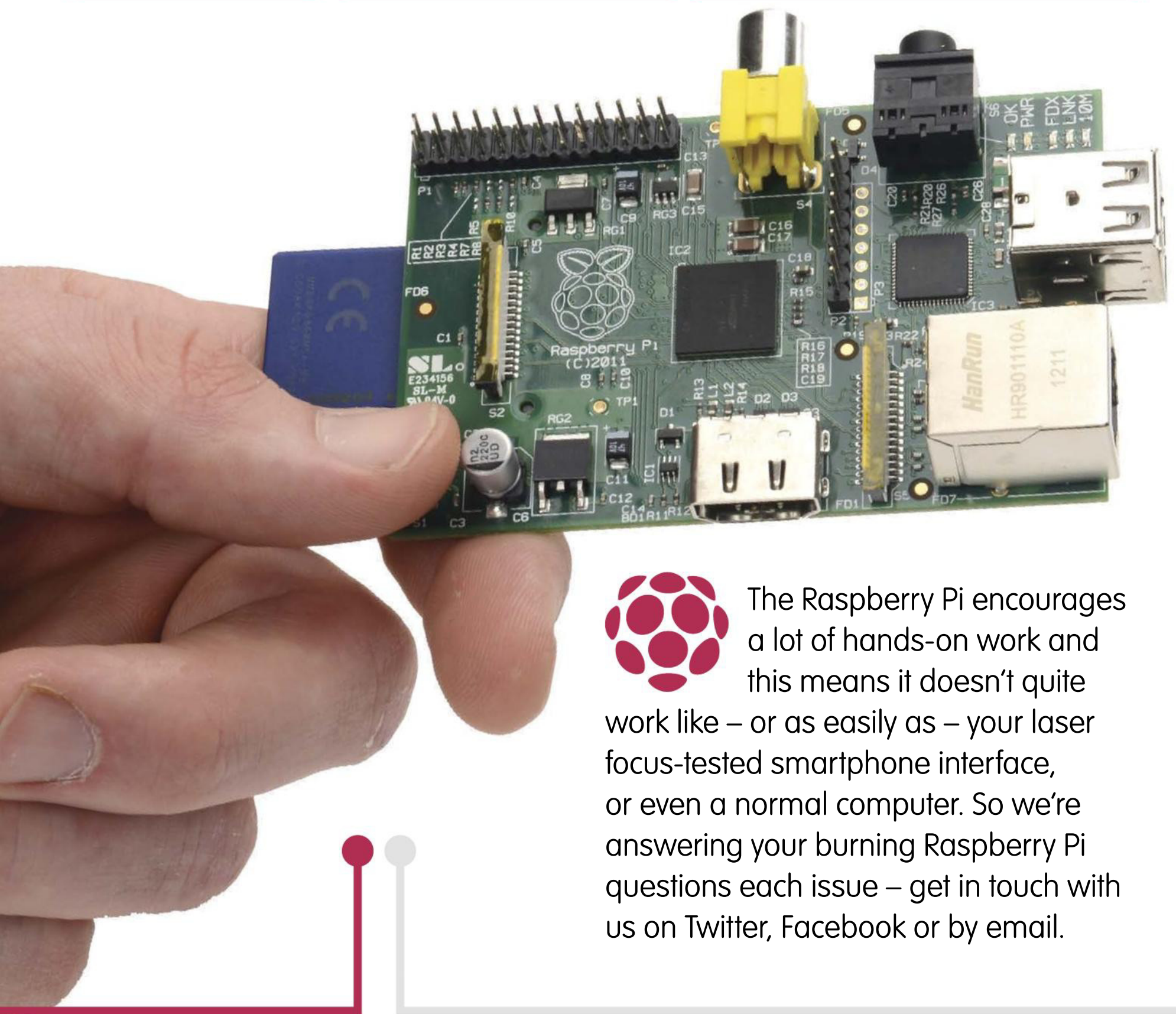
@linuxusermag



Linux User & Developer



RasPi@imagine-publishing.co.uk



The Raspberry Pi encourages a lot of hands-on work and this means it doesn't quite work like – or as easily as – your laser focus-tested smartphone interface, or even a normal computer. So we're answering your burning Raspberry Pi questions each issue – get in touch with us on Twitter, Facebook or by email.

I hear a lot about the Pi being used in 'Internet of Things' projects, but what does the term actually mean?
George via email

The Internet of Things is an odd term – after all, everything is, on some level, a thing – but what it refers to in this context is the slew of smart and semi-smart devices that talk to each other on the internet and are designed to automate or allow us to easily control aspects of our day-to-day life. For example,

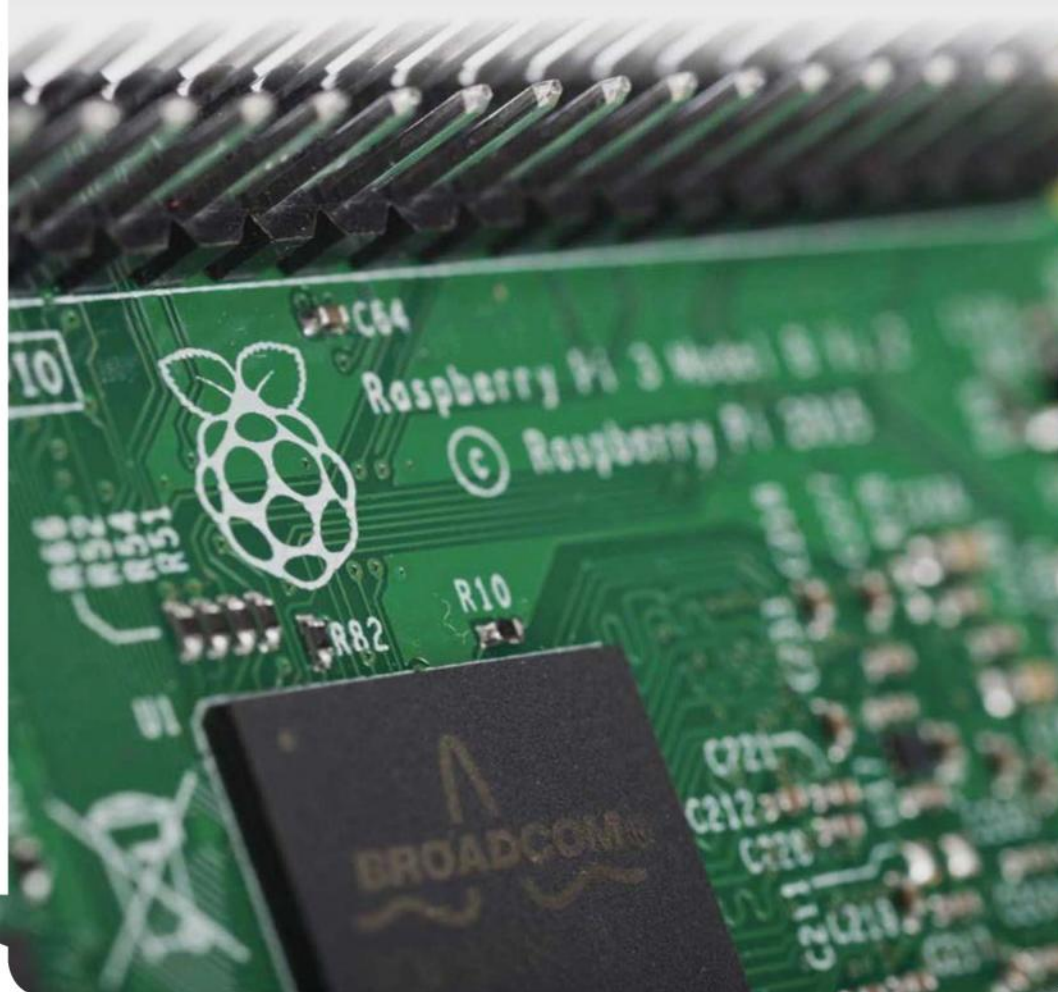
some of the most common Internet of Things devices are smart thermostats, which turn your heating up and down as they monitor temperatures. They can also be accessed via an app so you can, for example, turn up the temperature on your way home. The Raspberry Pi's size and power makes it perfect for inclusion in devices like this.



Keep up with the latest Raspberry Pi news by following @LinuxUserMag on Twitter. Search for the hashtag #RasPiMag

JUST A SCORE
WHAT'S YOUR JUST A SCORE?

Have you heard of Just A Score? It's a new, completely free app that gives you all the latest review scores. You can score anything in the world, like and share scores, follow scorers for your favourite topics and much more. And it's really good fun!



My Pi runs out of juice when I power it via USB from my PC.

Ben via email

Stop right there! While it's tempting to power your Pi from your PC, especially when you're working across the two devices, it's something you should never do! Just because its power

connector is USB doesn't mean that you can daisy chain it with another device – you'll soon run out of power and possibly damage your Pi. Please only ever use mains power as the power source for your Pi, otherwise you risk damaging it or improperly shutting down your SD card and losing data.



My display is upside-down on my touchscreen when in the case!

Tomaz via email

We've heard of this odd little quirk happening before and luckily there's a fairly simple fix for it. It's a simple configuration error in which the Pi doesn't take into account which way round it

is. To fix it, you need to edit a file called `sudo vi /boot/config.txt`. Open it up and add this line to the bottom of the file: `lcd_rotate=2`. The bottom of the file should now look something like this and the screen should now be the 'right' way round when it's in the case:

```
# Correct display orientation  
lcd_rotate=2
```



**JUSTA
SCORE**
WHAT'S YOUR JUST A SCORE?

You can score absolutely anything on Just A Score. We love to keep an eye on free/libre software to see what you think is worth downloading...

10 LinuxUserMag scored 10 for
Keybase

9 LinuxUserMag scored 9 for
Cinnamon Desktop

8 LinuxUserMag scored 8 for
Tomahawk

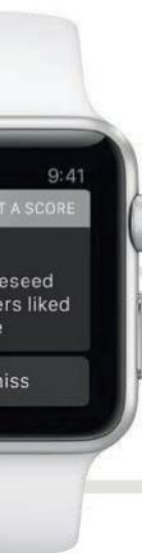
4 LinuxUserMag scored 4 for
Anaconda installer

3 LinuxUserMag scored 3 for
FOSS That Hasn't Been
Maintained In Years

SCORE ANYTHING
JUST A SCORE



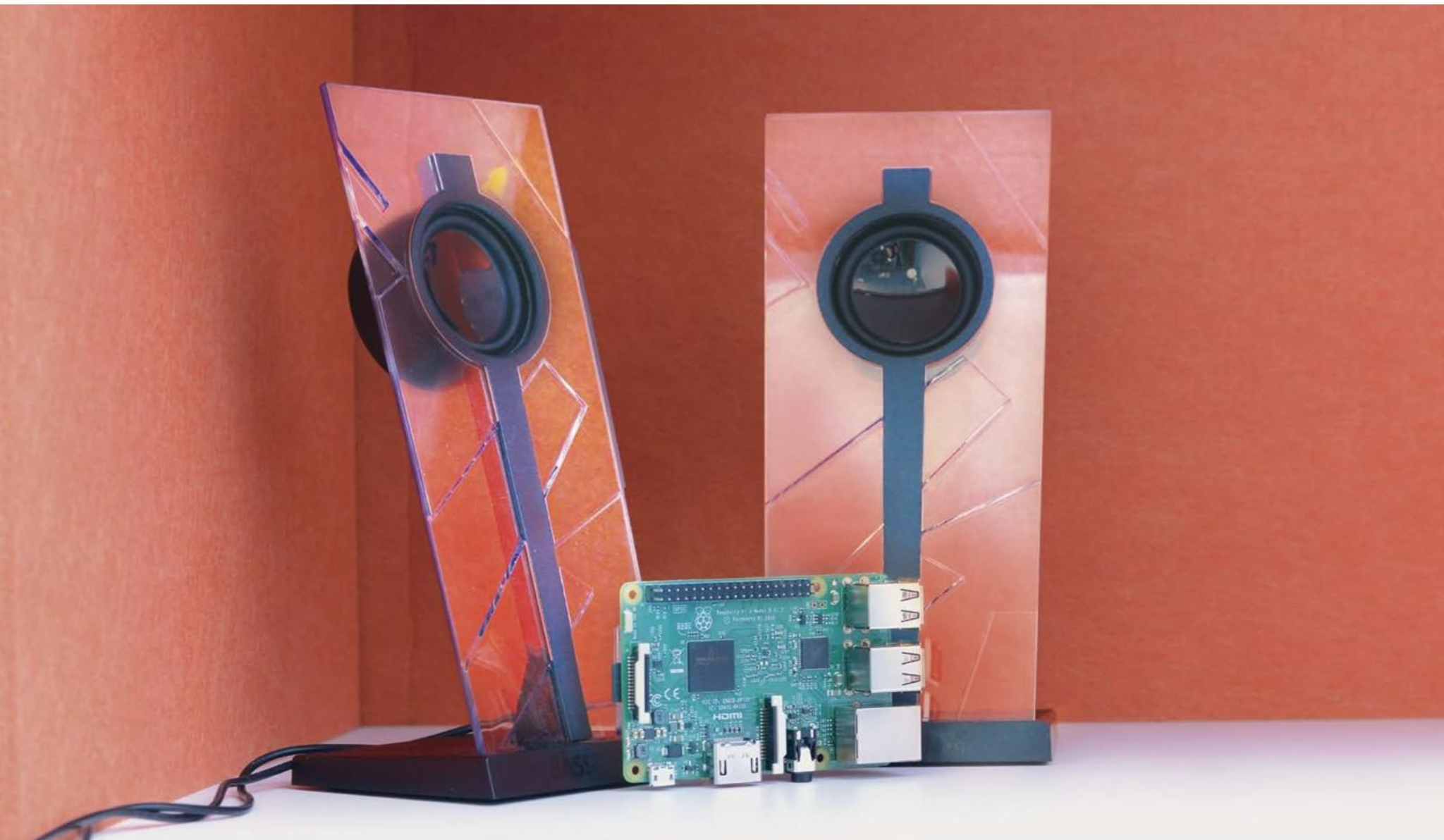
Download on the
App Store





Next issue

 Get inspired  Expert advice  Easy-to-follow guides



Set up a multi-room sound system

Get this issue's source code at:
www.linuxuser.co.uk/raspicode